

# Mikrokontrolery?

## To takie proste...

W poprzednim odcinku zapoznaliśmy się z droższymi Czytelnicy z prostymi układami I/O, dzięki którym możliwe jest sterowanie zewnętrznymi urządzeniami za pomocą mikrokomputerka edukacyjnego.

W przykładach wykorzystałem najpopularniejsze układy serii TTL (HCT-TTL), toteż zastosowanie ich w praktycznych układach nie powinno nasręcać problemów.

Tym razem mam zamiar przedstawić bardziej rozbudowane układy, umożliwiające np. drukowanie na typowej drukarce komputerowej, bądź wykorzystanie szybkiej transmisji równoległej z urządzenia zewnętrznego do komputerka edukacyjnego.

### Część 18 Programowane układy wejścia-wyjścia

Układ który mam zamiar dzisiaj przedstawić jest tak stary jak cała technika komputerowa, oczywiście ta z prawdziwego zdarzenia, czyli z przełomu lat 70-tych i 80-tych. Mowa będzie o scalaku oznaczanym przez producentów jako 8255 – programowany układ wejścia/wyjścia z łączem równoległym. Zanim jednak przejdę do części praktycznej i przedstawię konkretne schematy oraz listingi procedur obsługi układu 8255 wypada mi krótko przedstawić sam układ, tak aby nikt nie czuł niedosytu wiedzy w tym zakresie. Jeżeli zaś któryś z Was drodzy Czytelnicy jest na tyle niecierpliwy, że nie może się powstrzymać od skonfrontowania wiedzy praktycznej w rzeczywistości, może przejść od razu do akapitu "Drukowanie z komputerka".

### 8255 – Dużo funkcji niewielkim kosztem

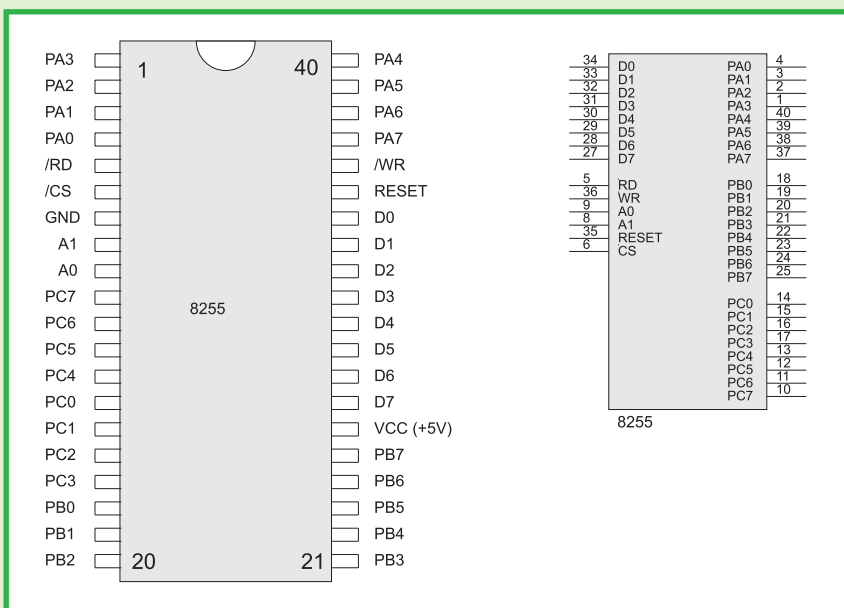
Bohater naszego dzisiejszego odcinka to jegomość umieszczony w 40-nóżkowej typowej obudowie DIL. Dzięki temu zastosowania i montaż tego układu na płytce drukowanej, podobnie jak procesorów 8051/2 nie nasręca trudności. Ma to szczególnie znaczenie bo w dzisiejszej erze miniaturyzacji, na rynku rozpanoszyły się wszelkiego rodzaju zminiaturyzowane obudowy, które niejednego amatora elektronika mogą przyprawić o zawrót głowy.

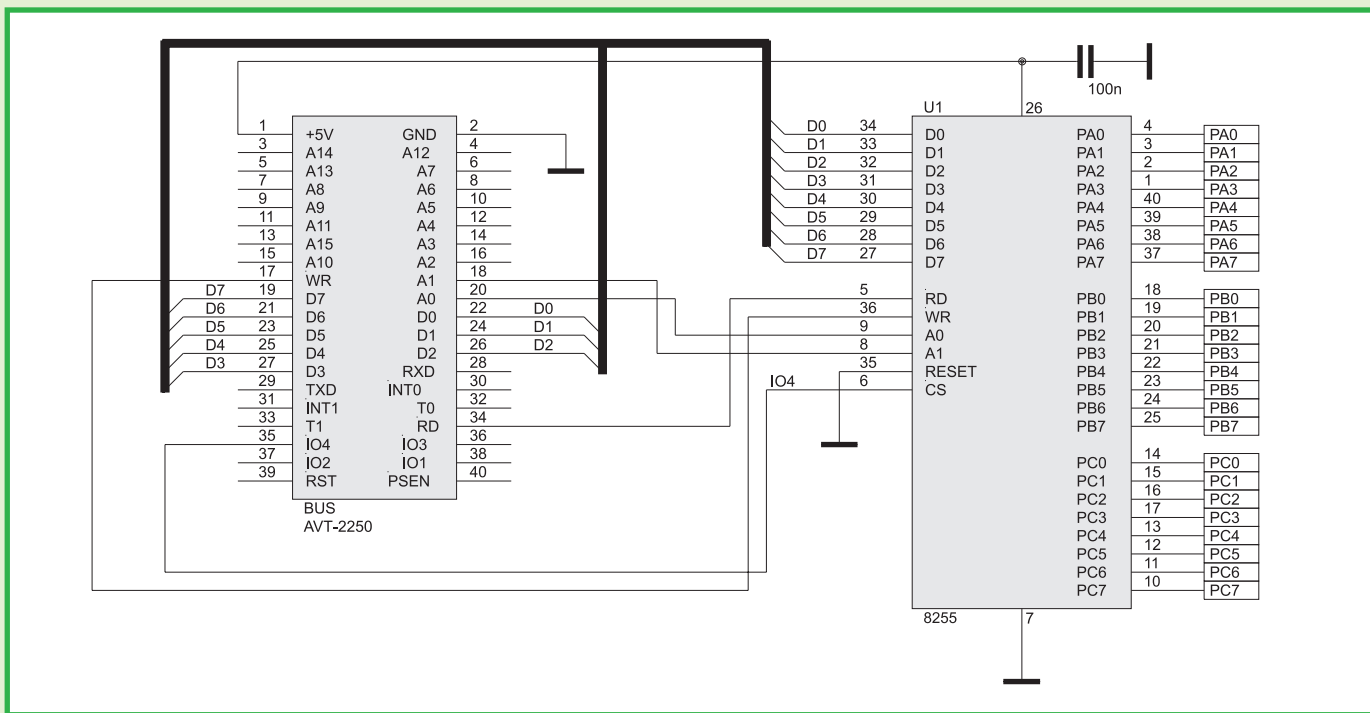
W czasach, kiedy technika komputerowa PC wkraczała pod strzechy, kiedy to IBM wyprodukował pierwszy komputer osobisty, a w chwilę później pojawiło się na rynku wielu innych producentów sprzętu, w układach elektronicznych komputerów PC stosowano układy 8255 bardzo często. Pierwszym zastosowaniem była obsługa łącza równoległego standardu Centronics, przeznaczonego pierwotnie dla potrzeb drukowania. W chwili obecnej kiedy wszystkie układy I/O komputera PC znalazły się na "płytcie głównej", układ 8255 wraz z całą rodziną wcześniej produkowanych scalaków z mikroprocesorowej rodziny Intel'a znalazł swoje miejsce z raczej został "wcielony" do tzw. układów "chip set'u" płyty komputera.

Pomimo tego, że w dzisiejszych komputerach PC do sterowania portem równoległym kostki 8255 nie są już wykorzystywane, to jednak można je spotkać w przemysłowych wersjach komputerów sterujących - także PC. Stary, ale sprawdzony standard przyjął się tak dobrze jak poczciwa już pięćdziesiątka-jedynka i na razie nic nie zapowiada jej upadku. Układ 8255 jest nadal produkowany i można go nabyć prawie w każdym sklepie ze specjalistycznymi artykułami elektronicznymi. Koszt kostki mieszczący się na poziomie około 5..10 zł nie jest kwotą wygórowaną, a pamiętając że układ za tak niską cenę oferuje wiele możliwości, dostajemy go praktycznie za darmo.

Najprościej rzecz mówiąc, układ 8255 zawiera trzy uniwersalne 8-bitowe porty (PA, PB i PC), z których każdy może być skonfigurowany jako wejście, wyjście a w pewnych trybach spełniać rolę mieszaną, nawet z możliwością generowania przerwań do procesora. Dodatkowo, oprócz wspomnianych

Rys.1 Topografia wyprowadzeń układu 8255 i jego schemat funkcjonalny





Rys.2 Sposób dołączenia kostki 8255 do komputerka edukacyjnego.

rejestrów portów PA, PB i PC układu 8255 zawiera tzw. czwarty rejestr – konfiguracyjny, dzięki któremu możliwe jest ustalenie, jak ma pracować cały układ, czy np. port PA ma być ustawiony jako wejściowy, czy wyjściowy itp.

Dołączenie układu do komputerka edukacyjnego (generalnie do procesorów 8051/2) jest bardzo proste, bowiem 8255 sterowany jest sygnałami kompatybilnymi z tymi generowanymi przez mikrokontrolery serii MCS-51. Sposób dołączenia kostki do komputerka edukacyjnego omówię za chwilę. Tymczasem na rys.1 przedstawiam rozkład wyprowadzeń kostki 8255.

Jak widać do połączenia układu 8255 z urządzeniami zewnętrznymi służą linie PA0...PA7, PB0...PB7 i PC0...PC7. To w jaki sposób ustawione są te linie oraz ich funkcje, determinuje specjalne słowo (bajt) konfiguracyjne (wspomniany czwarty rejestr) - zaszyte wewnątrz układu 8255. Jak te słowo zapisywać, innymi słowy jak konfigurować cały układ powiem za chwilę. Na razie wyjaśnijmy sobie znaczenie poszczególnych wyprowadzeń układu. Co prawda jest ich aż 40-ci ale jak się za chwilę przekonasz, drogi Czytelniku, nie jest to wcale dużo.

**D0...D7** – linie dołączane do szyny danych systemu mikroprocesorowego. Dzięki nim, możliwa jest transmisja danych z procesora nadzorującego do i z układu 8255. W przypadku komputerka edukacyjnego AVT-2250 linie te powinny dołączyć się do szyny adresowej D0...D7 komputerka – wyprowadzenia: 22,24,26,27,25,23,21,19 złącza BUS AVT-2250

**/CS** – sygnał wejściowy wyboru – selekcji układu przez procesor. Podanie logicznego zera na to wejście umożliwia transmisję danych poprzez linie D0...D7. W układzie komputerka edukacyjnego sygnał ten może być dołączony do jednego z wyjść dekodera adresowego IO3 lub IO4, tak jak w przykładach z poprzedniego odcinka klasy mikroprocesorowej.

**/WR** – sygnał zapisu danych do układu 8255 przez procesor zewnętrzny. Podanie stanu niskiego na te wejście powoduje zapisanie danych z szyny D0...D7 do wewnętrznych rejestrów układu 8255. W układzie komputerka edukacyjnego sygnał ten łączy się bezpośrednio z końcówką /WR procesora (lub pinem 17 złącza BUS na płytce głównej AVT-2250).

**/RD** – sygnał odczytu danych z układu 8255. Podanie stanu niskiego na te wejście spowoduje odczytanie informacji z wewnętrznych rejestrów układu 8255. Podobnie jak w przypadku sygnału zapisu, wejście to powinno się dołączyć do wyjścia /RD procesora 8051 (lub do pinu 34 złącza BUS komputerka AVT-2250).

**A1, A0** – linie sterujące wyborem jednego z czterech rejestrów wewnętrznych układu 8255. W zależności od poziomów na tych liniach podczas cyklu odczytu/zapisu przez procesor wybrany zostaje jeden z trzech rejestrów wyjściowych PA, PB, PC bądź rejestr konfiguracyjny.

**RESET** – wejście zerowania układu 8255. Podanie stanu wysokiego na to wejście spowoduje wyzerowanie rejestrów wewnętrznych

(rejestr sterującego) układu 8255 i ustawienie portów PA, PB, PC jako wejściowych. Taki sam stan układu 8255 ustawiany jest automatycznie po każdorazowym włączeniu zasilania.

**PA0...PA7** – linie 8-bitowego uniwersalnego portu PA (pierwszego portu)

**PB0...PB7** – j/w lecz portu PB (drugiego portu)

**PC0...PC7** – j/w lecz portu PC (trzeciego portu)

Na rys.2 przedstawiony jest najprostsz sposób dołączenia układu 8255 do komputerka AVT-2250. Jeżeli śledzisz uważnie cykl klasy mikroprocesorowej, w szczególności jeśli zapoznałeś się z poprzednim odcinkiem o prostych układach wejścia/wyjścia, z pewnością stwierdzisz, że sposób połączenia jest oczywisty i analogiczny z układami z poprzedniego odcinka klasy.

I tak linie danych D0...D7 dołączamy do szyny danych komputerka BUS – D0...D7. Linie sterujące zapisem /WR i odczytem /RD do odpowiednich linii złącza BUS (/WR i /RD). Sygnał wyboru układu /CS dołączamy bezpośrednio do wyjścia dekodera adresowego w układzie komputerka np. IO4, natomiast linie wyboru rejestru do analogicznych linii adresowych szyny adresowej komputerka A0 i A1. Prawda że proste! I to już wszystko, moi drodzy, można zająć się programowaniem układu. Aha byłby zapomniał o konieczności dołączenia zasilania, które przyłącza się trochę nietypowo (uwaga dla tych którzy w tym momencie jedną ręką i jednym okiem chcą natychmiast zabrać się za wykonanie płytki drukowanej), mianowicie +5V dołącza się do końcówki 26 (a nie jak typowo w obudowach DIL do pinu 40) a masę do końcówki 7.

Projektując płytkę lub umieszczając układ na "uniwersalce" warto zablokować linię zasilającą tuż przy układzie 8255 kondensatorem 100nF.

W ten sposób po dołączeniu układu do komputerka mamy do dyspozycji trzy 8-mio bitowe porty PA, PB i PC, do których możemy dołączyć dowolne urządzenia.

Wnikliwy czytelnik z pewnością zauważy konsekwencję z zastosowania dwóch linii adresowych A0 i A1 w układzie 8255 ze względu na sterowanie 4-roma rejestrami wewnętrznymi układu 8255. Tabela 1

Tabela 1

A1	A0	wybrany rejestr 8255
0	0	rejestr portu PA
0	1	rejestr portu PB
1	0	rejestr portu PC
1	1	rejestr kontrolny (CTRL) – konfiguracji układu

przedstawia znaczenie sygnałów A0 i A1 w odniesieniu do wyboru jednego z czterech rejestrów wewnętrznych kostki 8255.

W połączeniu ze sterowaniem sygnałem wyboru /CS poprzez linię np. IO4 poszczególne rejestry 8255 będą widziane przez procesor komputera jako adresy:

- PA – adres A000h (A1,A0 = 0,0)
- PB – adres A001h (A1,A0 = 0,1)
- PC – adres A002h (A1,A0 = 1,0)
- CTRL – adres A003h (A1,A0 = 1,1)

Analogicznie, jeśli podłączylibyśmy linię wyboru /CS np. do sygnału IO3 komputera to wtedy jak się zapewne domyślasz, adresy wyglądałyby odpowiednio tak: 8000h, 8001h, 8002h, 8003h.

Toteż, jeżeli po włączeniu zasilania komputerka z dołączonym do niego układem 8255, procesor (51-ka) wykona instrukcję:

```
MOV DPTR, #A000h ; adres rejestru portu PA
MOVX A, @DPTR ; odczyt rejestru PA z 8255
```

to w efekcie w akumulatorze znajdzie się liczba której poszczególne bity będą zgodne z poziomami sygnałów na liniach PA0...PA7 układu 8255. Podobnie można będzie odczytać linie PB0...PB7 w następujący sposób:

```
MOV DPTR, #A001h
MOV A, @DPTR
```

i rejestru C

```
MOV DPTR, #A002h
MOV A, @DPTR
```

O ile rejestr PA (a także PB) może pełnić rolę wejść lub wyjść cyfrowych (nie jest możliwe aby niektóre bity tych rejestrów były wejściami a inne wejściami jednocześnie – w tej samej chwili), o tyle rejestr PC ma szczególne dodatkowe właściwości. Otóż może on być tak skonfigurowany (za pomocą rejestru kontrolnego CTRL), że w jednocześnie z portu PC można odczytywać dane lub je zapisywać. Oczywiście w takim przypadku wybrane linie PC będą pracowały wyłącznie jako wejścia, pozostałe zaś jako wyjścia. Fizycznie w układzie 8255 w odróżnieniu do rejestrów PA i PB, które widziane są jako całe 8-mio bitowe rejestry, rejestr PC może być modyfikowany w dwóch "połówkach". I tak starsze cztery bity rejestru PC7...PC4 będą oznaczane jako PCa młodsze PC3...PC0 jako PCb. Obrazowo wszystkie 3 rejestry portów przedstawiam poniżej.

### Rejestr PA - końcówki 1-4 i 37-40 układu 8255

nr bitu:	7	6	5	4	3	2	1	0	
bity	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	<b>PA</b>

### Rejestr PB - końcówki 18-25 układu 8255

nr bitu:	7	6	5	4	3	2	1	0	
bity	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	<b>PB</b>

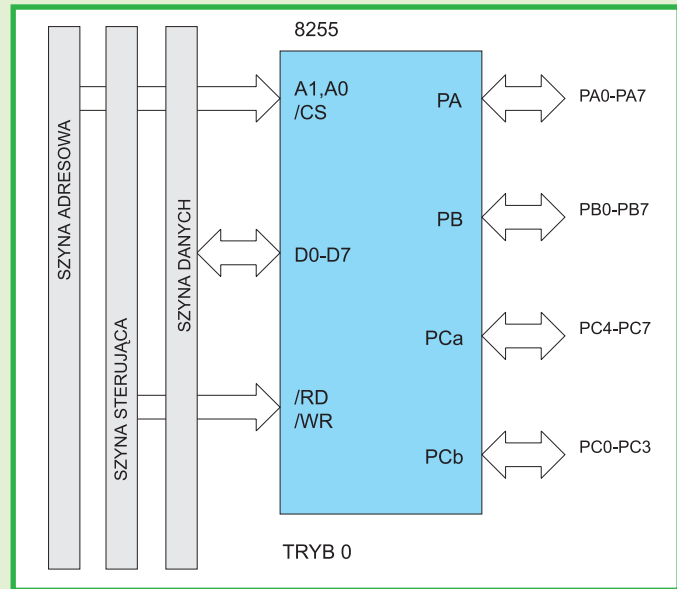
### Rejestr PC - końcówki 10-17 układu 8255

nr bitu:	7	6	5	4	3	2	1	0	
bity	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	<b>PC</b>
	rejestr PCa				rejestr PCb				

Ważną informacją jest to że odczyt rejestru kontrolnego CTRL (pod adresem A003h) jest, uwaga, zabroniony! Rejestr ten można tylko zapisywać. A czym go zapisywać? Otóż zapoznamy się teraz z budową tego rejestru i jego bitami oraz ich znaczeniem dla pracy kostki 8255.

### Rejestr kontrolny 8255 (CTRL) w trybie konfigurowania

nr bitu:	7	6	5	4	3	2	1	0	
bity	1	D6	D5	D4	D3	D2	D1	D0	<b>CTRL</b>



Rys.3 Układ 8255 w trybie pracy 0

W trybie programowania układu najstarszy bit (7) rejestru powinien być zawsze ustawiony ("1"). Znaczenie pozostałych bitów jest następujące:

bit 7 – w trybie programowania układu 8255 zawsze "1".  
bity D6 i D5 – określają wybór trybu pracy układu i rejestrów PA oraz PCa, i tak:

- 00 = tryb 0
- 01 = tryb 1
- 10 lub 11 = tryb 2. Znaczenie poszczególnych trybów omówię za chwilę.

bit D4 – określa kierunek pracy linii portu PA i tak

- 1 = PA jako wejście
- 0 = PA jako wyjście cyfrowe

W trybie 2 pracy układu 8255, bit ten nie ma znaczenia.

bit D3 – określa kierunek pracy linii portu PCa (PC7...PC4) i tak

- 1 = PCa jako wejście
- 0 = PCa jako wyjście cyfrowe

bit D2 – określa wybór trybu pracy rejestru PB, i tak:

- 0 = tryb 0
- 1 = tryb 1

bit D1 – określa kierunek pracy linii portu PB, i tak:

- 1 = PB jako wejście
- 0 = PB jako wyjście cyfrowe

bit D0 – określa kierunek pracy linii portu PCb (linie PC3...PC0) i tak

- 1 = PCb jako wejście
- 0 = PCb jako wyjście

Jak widać regułą jest, że ustawienie bitów D0, D1, D3, D4 powoduje ustawienie odpowiadających im portów jako wejścia, natomiast wyzerowanie tych bitów powoduje ustawienie tych portów jako wyjścia cyfrowe. W najprostszym trybie pracy mamy zatem aż 16 kombinacji ustawień portów PA, PB i dwóch połówek portu PC.

## TRYB 0

Wspomniane tryby pracy (0, 1 lub 2) umożliwiają wybór sposobu działania całego układu. Dla uproszczenia zajmijmy się na razie najprostszym trybem 0. W trybie tym jak powiedziałem wcześniej każdy z rejestrów PA, PB oraz dwóch połówek PC (PCa i PCb) może pracować jako wejście lub wyjście cyfrowe. Dane przesyłane są z szyny danych D0...D7 systemu mikroprocesorowego do rejestrów portu (ustaw-

# Też to potrafisz

bajt sterujący CTRL (binarnie)	zapis hex	port PA	port PB	port PC	
				linie PC7...PC4	linie PC3...PC0
100 1 1 0 1 1	9Dh	WE	WE	WE	WE
100 0 1 0 1 1	8Dh	WY	WE	WE	WE
100 1 1 0 0 1	99h	WE	WY	WE	WE
100 0 1 0 0 1	89h	WY	WY	WE	WE
100 1 0 0 1 1	93h	WE	WE	WY	WE
100 0 0 0 1 1	83h	WY	WE	WY	WE
100 1 0 0 0 1	91h	WE	WY	WY	WE
100 0 0 0 0 1	81h	WY	WY	WY	WE
100 1 1 0 1 0	9Ah	WE	WE	WE	WY
100 0 1 0 1 0	8Ah	WY	WE	WE	WY
100 1 1 0 0 0	98h	WE	WY	WE	WY
100 0 1 0 0 0	88h	WY	WY	WE	WY
100 1 0 0 1 0	92h	WE	WE	WY	WY
100 0 0 0 1 0	82h	WY	WE	WY	WY
100 1 0 0 0 0	90h	WE	WY	WY	WY
100 0 0 0 0 0	80h	WY	WY	WY	WY

**Tabela 2**

ionych jako wyjścia) lub odczytywane poprzez tą szynę do procesora w sposób niesynchronizowany.

Rys.3 obrazuje ideę pracy układu 8255 w trybie 0.

W przypadku kiedy procesor ustawi np. rejestr PA jako wyjście cyfrowe, to zapisując potem ten rejestr za pomocą instrukcji:

```
MOVX @DPTR,A
```

powoduje że dane z szyny danych D0...D7 pojawią się na liniach portu PA7...PA0, zgodnie z daną umieszczoną w akumulatorze Acc). W przypadku zaś odwrotnym, kiedy np. rejestr PA pracuje jako wejściowy, odczyt rejestru za pomocą instrukcji

```
MOVX A,@DPTR
```

spowoduje pojawienie się na liniach D0...D7 szyny danych (a potem w akumulatorze Acc) stanów linii portu PA7...PA0).

I tak dla przykładu zapiszmy słowo sterujące CTRL układu 8255 w taki sposób żeby np.

port PA i PC pracowały jako WYJŚCIA  
port PB jako WEJŚCIE,

patrzac na opis bitów rejestru (wyżej) zapiszemy:

D6, D5 i D2 będą = 0 (tryb 0)  
D4 = 0 to PA będzie wyjściem  
D3 = 0 to PCa – wyjście  
D1 = 1 to PB – wejście

wreszcie

D0 = 0 to PCb jako wyjście (podobnie jak PCa)

zatem aby tak skonfigurować układ 8255 należy wykonać sekwencję instrukcji:

```
MOV A, #1000010b ;bity D7, D6, D5, D4, D3, D2, D1, D0
MOV DPTR, #A003h ;zakładam, że /CS jest dopięty do /IO4 komputerka
MOVX @DPTR, A ;i zapisanie konfiguracji do układu 8255
```

**Rejestr kontrolny 8255 (CTRL) w trybie modyfikacji rejestru PC**

nr bitu:	7	6	5	4	3	2	1	0	
bity	1	D6	D5	D4	D3	D2	D1	D0	CTRL

Aby ułatwić konfigurowanie układu w trybie pracy 0 w praktycznych zastosowaniach, poniżej zamieszczam tabelę 2, w której zapisane są wszystkie wartości rejestru kontrolnego CTRL, które dają jeden z 16 sposobów ustawień portów PA, PB i PC.

W tabeli dla zwiększenia czytelności zapisałem wartość wpisywaną do rejestru kontrolnego CTRL w postaci binarnej oraz szesnastkowej. Dodatkowo rozdzieliłem spacjami niektóre bity w zapisie binarnym, tak abyś mógł łatwiej przeanalizować sposób ustawienia lub zerowania poszczególnych bitów zgodnie z wcześniejszym opisem.

Teraz jeżeli chcemy ustawić porty PA, PB i PC jako wyjścia (np. do sterowania 24-oma diodami LED) wystarczy sięgnąć do tabeli i odczytać liczbę 80h jako stosowną do zapisu do rejestru konfiguracyjnego układu 8255. Potem wystarczy wykonać ciąg instrukcji:

```
MOV A, #80h
MOV DPTR, #A003h
MOVX @DPTR, A ;i układ 8255 skonfigurowany
```

Teraz jeżeli chcemy np. "zapalić" linie portu PA, to wystarczy wydać polecenie

```
MOV A, #255
MOV DPTR, #A000h
MOVX @DPTR, A
```

W praktyce przy pisaniu programu źródłowego (komputerowcy) warto posłużyć się deklaracją EQU dla zdefiniowania poszczególnych portów układu 8255. Wtedy nie trzeba będzie za każdym razem pamiętać, co jakie adresy mają porty PA, PB czy PC. W naszym przykładzie można to zrobić w sposób następujący:

```
IO_PA EQU A000h ;deklaracja adresu portu PA
IO_PB EQU A001h ;i/w lecz portu PB
IO_PC EQU A002h ;i/w lecz portu PC
IO_CTRL EQU A003h ;wreszcie rejestru kontrolnego 8255
```

Teraz wystarczy odpowiednio i elegancko ich używać, np. tak:

```
MOV A, #91h ;ustawiam: PA i PCb jako WE, PB i PCa jako WY
MOV DPTR, #IO_CTRL ;adres rejestru kontrolnego
MOVX @DPTR, A ;i skonfigurowanie układu

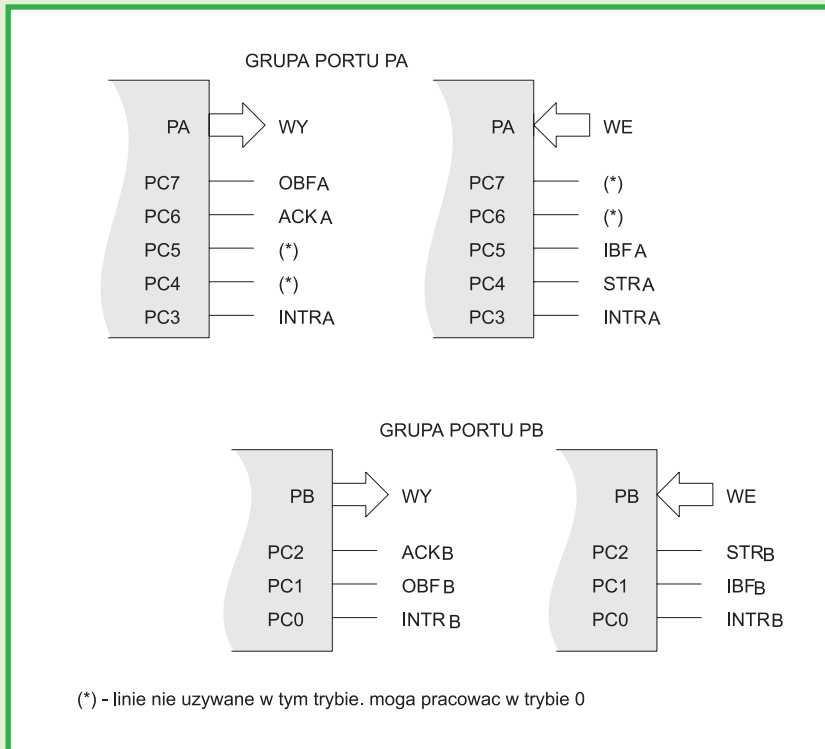
MOV DPTR, #IO_PA ;adres rejestru PA
MOVX A, @DPTR ;i odczyt linii PA7...PA0

MOV DPTR, #IO_PB ;adres rejestru PB
MOV A, #55h ;ustaw naprzemiennie linie PB7...PB0 jako "01010101"
MOVX @DPTR, A ;i zapis do rejestru PB
..... ;itd.
```

Na szczególną uwagę zasługuje rejestr PC układu 8255. Otóż możliwe jest selektywne ustawianie bądź zerowanie poszczególnych bitów tego rejestru. Do tego celu służy zapis do rejestru kontrolnego słowa z wyzerowanym najstarszym bitem D7. Poniżej przedstawiam strukturę słowa rejestru CTRL dla takiego przypadku sterowania.

W trybie tym najstarszy bit (7) rejestru powinien być wyzerowany. Znaczenie pozostałych bitów jest następujące:

bit 7 – zawsze "0".  
bity D6, D5, D4 – nie używane (dowolna wartość, zazwyczaj "0")



Rys.4 Znaczenie linii portów PA, PB i PC w trybie pracy 1.

bit D3, D2, D1 – określają numer ustawianego lub zerowanego bitu rejestru PC, i tak:

- kolejność "000" oznacza wybór bitu D0 (linia PC0)
  - kolejność "001" oznacza wybór bitu D1 (linia PC1)
  - kolejność "010" oznacza wybór bitu D2 (linia PC2)
  - kolejność "011" oznacza wybór bitu D3 (linia PC3)
  - kolejność "100" oznacza wybór bitu D4 (linia PC4)
  - kolejność "101" oznacza wybór bitu D5 (linia PC5)
  - kolejność "110" oznacza wybór bitu D6 (linia PC6)
  - kolejność "111" oznacza wybór bitu D7 (linia PC7)
- bit D0 – powinien zawierać "1" jeżeli chcemy wybrany bit (linię portu) rejestru PC ustawić lub "0" - jeżeli wyzerować

Przykład: zapisaliśmy wcześniej w rejestrze PC następujące stany: "10101010", teraz chcemy wyzerować także linię PC7. należy więc wydać instrukcję:

```
MOV DPTR, #IO_CTRL
MOV A, #00001110b ;zeruj linię PC7
MOX @DPTR, A ;zapis do rejestru CTRL 8255
```

i gotowe! Selektowna modyfikacja bitów rejestru C możliwa jest tylko w trybie 0 pracy rejestrów PA lub PB. Jeżeli np. rejestr PA pracuje w trybie 1 lub 2, a rejestr PB w trybie 0 to można modyfikować jedynie bity rejestru PCb (z grupy B) i odwrotnie, dla PB pracującego w trybie 1, a PA w trybie 0, modyfikowalne są tylko bity PCa.

Tryby 1 i 2 pracy układu 8255 nie są tak proste jak tryb 0. Dzięki nim jednak możliwa jest synchronizacja transmisji pomiędzy układem 8255 a urządzeniem zewnętrznym. Można powiedzieć, że w praktyce trybu 0 używa się kiedy po prostu chcemy sterować zewnętrznymi urządzeniami dołączonymi (poprzez układy pośredniczące) do linii portów PA, PC czy PB, lub także odczytywać stany tych linii w dowolnym momencie. Istotną informacją jest to że w trybie 0 informacja zapisana do każdego z rejestrów portów PA, PB, PC jest zapamiętywana do kolejnej modyfikacji wybranego rejestru. Natomiast w przypadku odczytu, rejestr danego portu jest przezroczysty dla danych przesyłanych z urządzenia zewnętrznego i nie zapamiętuje ich.

Natomiast tryby 1 i 2 używane są do bardziej złożonych zastosowań, takich właśnie jak transmisja pomiędzy inteligentnymi układami peryferyjnymi, takimi jak np. drukarka, komputer, inny układ oparty o 8255 lub podobny. Zagadnienie jest bardzo rozbudowane, dlatego poniżej skupię się do skróconego omówienia sposobu działania układu 8255 w tych trybach. Przedstawione zaś w dalszej części artykułu przykłady praktycznego zastosowania kostki 8255 w celach współpracy ze zwykłą drukarką komputerową oraz do transmisji danych z innego urządzenia

zewnętrznego, pozwolą na zilustrowanie tego zagadnienia w sposób wystarczający do naszych zastosowań w przyszłych wspólnie omawianych układach.

## TRYB 1

W tym trybie pracy dane mogą być przesyłane do i z innych urządzeń poprzez porty PA lub PB. W tym trybie transmisja jest jednokierunkowa, tzn. dane mogą być odczytywane przez cały czas do kolejnej rekonfiguracji układu 8255 lub wyłącznie zapisywane. Niektóre z linii portu PC są w tym trybie przyporządkowane rejestrówi PA lub PB, a służą do przesyłania sygnałów synchronizujących transmisję, między rejestrami PA lub PB a urządzeniem zewnętrznym. Rejestr PA z odpowiednimi wybranymi liniami portu PC (PCa), które synchronizują transmisję nazywa się grupą A, natomiast rejestr PB z pozostałymi liniami portu PC (PCb) – grupą B.

Tryby pracy linii grupy A i grupy B jest programowany niezależnie. Toteż gdy jeden z rejestrów PA lub PB wraz z odpowiednimi liniami portu PC może pracować w trybie 1 to tryb pracy drugiego i linii PC należącymi wraz z nimi do grupy może pracować w trybie 1 lub 0. Rys.4 pokazuje jak układ 8255 pracuje w trybie 1.

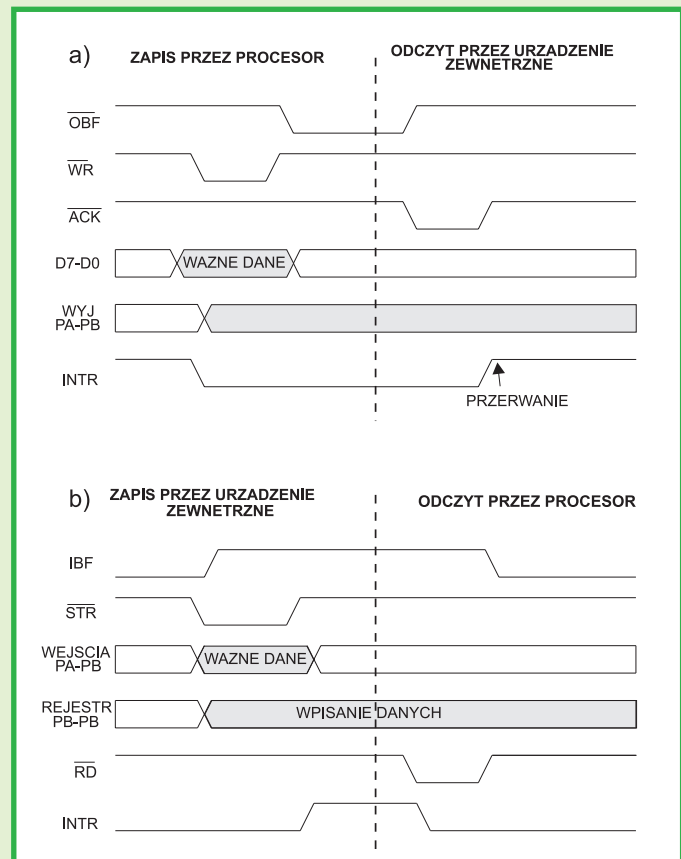
Jak wspomniałem port PC generuje na liniach dodatkowe sygnały synchronizujące transmisję danych pomiędzy 8255 a urządzeniem zewnętrznym. znaczenie sygnałów tych jest następujące:

a) przy transmisji z układu 8255 do urządzenia zewnętrznego

**/OBF** – (ang. "output buffer full") sygnał ten przyjmuje poziom niski podczas całego cyklu przyjęcia danej z procesora (od narastającego zbocza sygnału /WR) do odebrania danej przez układ zewnętrzny (który sygnalizuje potwierdzenie odebrania znaku na linii /ACK)

**/ACK** – (ang. "acknowledge") urządzenie zewnętrzne po odebraniu znaku z portu (np. PA) potwierdza zakończenie odczytu wystawiając poziom niski na tej linii, co 8255 odbiera jako zakończenie transmisji.

Rys.5 Zależności czasowe sygnałów synchronizacji transmisji podczas odczytu danej (a) przez urządzenie zewnętrzne oraz zapisu (b) danej do układu 8255 przez te urządzenie



## Też to potrafisz

**/INTR A,B** – (ang. “interrupt”) sygnały wyjściowe 8255 zgłoszenia przerwania. Przejście tego sygnału do stanu aktywnego oznacza że urządzenie zewnętrzne odczytała przesłaną daną i jest gotowe do przyjęcia kolejnej danej. Powrót do stanu nieaktywnego następuje wtedy kiedy procesor dokona ponownie zapisu tego słowa.

b) przy transmisji danej z urządzenia zewnętrznego do 8255

**IBF** – (ang. “input buffer full”) sygnał wyjściowy który przyjmuje stan aktywny w momencie kiedy dana zostaje zapisana przez urządzenie zewnętrzne w 8255, skasowanie tego sygnału następuje kiedy procesor odczyta z 8255 tak odebraną daną.

**/STR** – (ang. “strobe”) sygnał wejściowy, dzięki któremu urządzenie zewnętrzne może zgłosić gotowość do odebrania przez 8255 danej, wystawionej przez nie

**INTR A,B** – (ang. “interrupt”) sygnały wyjściowe 8255 zgłoszenia przerwania. Przejście tego sygnału do stanu aktywnego oznacza że urządzenie zewnętrzne zakończyło zapis danej w rejestrze wejściowym 8255. Powrót do stanu nieaktywnego następuje wtedy kiedy procesor dokona odczytu tej danej.

Na rys.5 a i b przedstawiono zależności czasowe odpowiednich sygnałów podczas transmisji w trybie 1. Analiza przebiegów szczególnie sygnałów sterujących nie jest trudna, należy jedynie skupić się nieco nad ideą pracy portu PC.

I tak w trybie przesyłania danej z 8255 do urządzenia zewnętrznego, procesor sterujący (u nas 8051) musi zapisać daną do układu 8255. Sygnał /WR przyjmie podczas zapisu stan niski. Równocześnie z momentem rozpoczęcie zapisu uaktywniony zostaje sygnał zgłoszenia przerwania INTR (poziom niski). Po zakończeniu zapisu danej przez procesor sygnał OBF przyjmuje stan niski, układ 8255 czeka na odebranie informacji przez urządzenie zewnętrzne. W tym czasie gotowa do odebrania dana znajduje się na wyprowadzeniach portu PA (lub PB – w zależności z której grupy A, czy B rejestrów korzystamy). Kiedy urządzenie zewnętrzne odczyta daną z portu PA (PB) zasygnalizuje ten fakt ustawieniem poziomu niskiego na linii /ACK co układ 8255 odbierze jako znak o zakończeniu odczytu daje przez te urządzenie. Sygnał /INTR zgłoszenia przerwania przyjmie stan nieaktywny (wysoki). Cykl transmisji został zakończony.

Podobnie sytuacja przedstawia się podczas transmisji znaku z urządzenia zewnętrznego do układu 8255. W tym przypadku urządzenie zewnętrzne po wystawieniu danej na port PA (PB) informuje układ 8255 o tym fakcie wystawiając poziom niski na linii /STR strobowania. W tym momencie dana ta zostaje zatrzaskowana w rejestrze PA (PB) a procesor dowiaduje się o tym otrzymując np. zgłoszenie przerwania poprzez wystąpienie stanu wysokiego na wyjściu INTR 8255. Następnie procesor odczytuje daną z portu PA (PB) poprzez zaadresowania portu i podanie stanu niskiego na linii /RD. Po odczycie transmisja zostaje zakończona, sygnał IBF przyjmuje stan nieaktywny, skasowane zostaje także przerwania INTR.

## TRYB 2

W trybie tym układ 8255 potrafi transmitować dane w dwóch kierunkach jednocześnie wraz z potwierdzeniem transmisji. Rejestr PA wykorzystywany jest wtedy do transmisji danych między 8255 a urządzeniem zewnętrznym. Jako sygnały synchronizujące wykorzysty-

wane są linie PC3...PC7. Pozostałe linie PC0...PC2 oraz rejestr PB mogą pracować niezależnie w trybie 0 lub 1.

Rys.6 ilustruje znaczenie linii portów PA i PC3...PC7 w trybie pracy 2. Znaczenie sygnałów sterujących jest takie jak w przypadku trybu 1.

## PRZERWANIA Z 8255

Jak wspomniałem wcześniej układ 8255 ma możliwość generowania przerwania dla potrzeb współpracującego z nim procesora. Wiesz już że sygnały te generowane są na określonych liniach portu PC. Dodatkowo trzeba wiedzieć, że w trybie 1 każdemu z portów PA i PB przyporządkowany jest tzw. przerzutnik zezwolenia przerwania, dzięki któremu możliwe jest w ogóle wygenerowanie przerwania przez 8255. Mówiąc prościej jeżeli w swojej aplikacji masz zamiar wykorzystywać sygnały INTR, to musisz odpowiednio ustawić dany przerzutnik, a robi się to bardzo prosto, bowiem poprzez zapis odpowiedniego bajtu do rejestru kontrolnego 8255. I tak dla rejestru PA jest to przerzutnik INTRA, a dla rejestru PB - INTRB.

Otóż aby ustawić dane przerzutniki i tym samym zezwolić na generowanie sygnału przerwania na linii INTR należy podobnie jak w przypadku (opisanym wcześniej) posłużyć się funkcją selektywnego ustawiania bitów rejestru PC.

W trybie 1 odpowiednie przerzutniki są przyporządkowane następującym bitom rejestru PC, i tak:

INTRB – bit D2 rejestru PC

INTRA (gdy port PA jest wejściem) – bit D4 rejestru PC

INTRA (gdy port PA jest wyjściem) – bit D6 rejestru PC

Dla przykładu podam że aby np. ustawić sygnał zezwolenia przerwania INTRB należy wykonać instrukcje:

```
MOV A, #0101b      ; ustawienie przerzutnika INTRB
MOV DPTR, #IO_PC   ; wybór rejestru PC
MOVX @DPTR, A      ; i zapis do niego
```

oczywiście nie można zapomnieć wcześniej skonfigurować układu 8255 do pracy rejestru PB w trybie 1, można to zrobić np. poleceniem:

```
MOV A, #10011111b
MOV DPTR, #IO_CTRL
MOVX @DPTR, A
```

W przykładzie ustawiłem bit D2 akumulatora który odpowiada za tryb pracy rejestru portu PB (1- tryb 1). Pozostałe porty ustawiłem domyślnie jako pracujące jako wejścia, a PA w trybie pracy 0.

W praktyce w układach często rezygnuje się z bezpośredniego połączenia linii INTR z wejściem przerwaniami procesora (w 8051 jest to INT0 lub INT1), ze względu na cenność wejść przerywających i możliwość monitorowania transmisji przez procesor poprzez tzw. “pulling”. Wtedy to wyjście INTR można podłączyć do jednej z linii wolnego portu 8255 pracującego w trybie 0. Cykliczny odczyt tej linii pozwoli na stwierdzenie czy transmisja została zakończona.

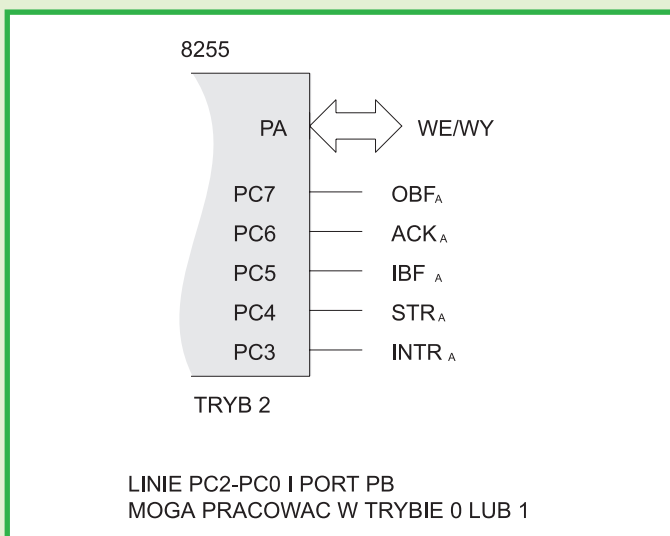
Oczywiście można stosować przerwania procesora, lecz jak wykazuje praktyka, operacje drukowania z małych systemów automatyki nie muszą być wykonywane w tle, chociaż nic nie stoi na przeszkodzie aby tak było.

No dobrze dość już teorii, zajmijmy się praktycznymi zastosowaniami. Poza oczywistym prostym zastosowaniem układu 8255 (tryb pracy 0) przedstawia poniżej dwa ciekawe zastosowania portu, pierwsze do obsługi drukarki, drugie do emulowania drukarki przez komputer, dzięki czemu możliwe jest transmitowanie danych z komputera wyposażonego w gniazdo zgodne ze standardem Centronics wprost do AVT-2250.

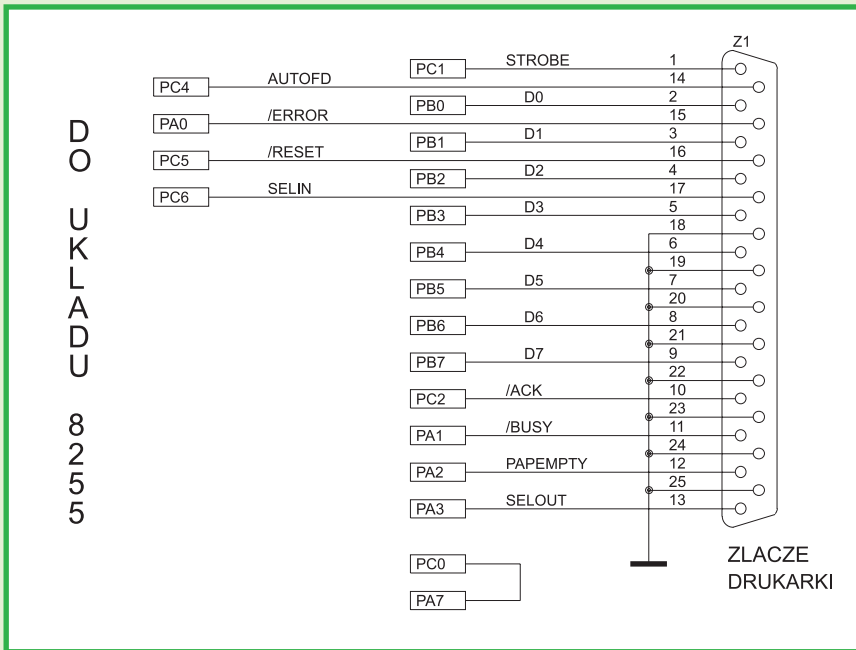
## DRUKOWANIE Z KOMPUTERKA AVT-2250

Nasz komputerek potrafi komunikować się za pomocą wyświetlacza LED oraz klawiatury lokalnej. Ciekawym uzupełnieniem, z pewnością także bardzo atrakcyjnym także dla zdrowszych o twój pierwszy system mikroprocesorowy, kolegów będzie możliwość obsługi drukarki. Będziesz mógł drukować co tylko przyjdzie Ci do głowy. Ja w przykładzie pokażę jak można uzupełnić program monitora komputerka AVT-2250 o polecenie drukowania zawartości pamięci operacyjnej w postaci szesnastkowej i znakowej. Funkcja ta może mieć niebanalne znaczenie szczególnie dla ręczniaków, którzy będą mogli “rzucić okiem” na swój cały program wydrukowany na kartce papieru. Oczywiście musicie moi drodzy posiadać choćby najstarszą zdezelowaną drukarkę. Ja posłużyłem się leciwą drukarką STAR LC-10, ach łezka kręci się w oku, kiedy wspominam tamte czasy – starych wolnych ale poczciwych pierwszych PC-tów.

Rys.6 Znaczenie sygnałów w trybie 2 pracy



Rys.7 Sposób podłączenia gniazda DB-25/F do układu 8255



Drukarkę taką można nabyć na giełdach w cenach przyporządkowanych o zawrót głowy, a mianowicie 50,- zł (tak, pięćdziesiąt złotych), mniej niż kosztuje sam komputer.

Przejdźmy zatem do układu.

Aby dołączyć drukarkę do naszego komputerka należy zaopatrzyć się w dodatkowe złącze typu DB-25/F (żeńskie). Następnie zgodnie z rys. 7 wykonujemy połączenia pomiędzy wyprowadzeniami układu 8255 (z rys.2) zgodnie z odpowiadającymi oznaczeniami na obu rysunkach.

Do pracy układu 8255 w charakterze sterownika drukarki wykorzystamy następujące ustawienia konfiguracji kostki 8255:

1. port PB ustawimy jako wyjściowy w trybie 1
2. port PA i PCa (PC4...PC7) ustawiamy w trybie 0, PA jako wejście, PCa - wyjście
2. wtedy sygnał na linii PC1 (/OBF) użyjemy jako strobojący dane do drukarki
3. stan na linii PC2 (/ACK) będzie sygnalizował układowi 8255 zakończenie odbioru znaku przez drukarkę
4. nie będziemy korzystać z układu przerwań procesora (przyda się być może do czego innego), dlatego wyjście zgłoszenia przerwania INTRB dołączymy zwrotnie do linii PA7 portu PA (będziemy badać stan tej linii aby wiedzieć kiedy transmisja jest zakończona) dodatkowe linie informacyjne z sygnałami wysłanymi przez drukarkę dołączymy do wolnych linii portu PA i PC, tak:
  - PA0 do linii /ERROR: sygnał na tej linii przyjmuje stan niski kiedy drukarka nie ma papieru, bądź jest w trybie off-line lub też po prostu jest wyłączona
  - PA1 do linii BUSY: wysoki poziom logiczny na tej linii sygnalizuje jedno ze zdarzeń:
    - a) drukarka jest w stanie off-line,
    - b) lub odłączona od komputera,
    - c) wewnętrzny bufor drukarki jest pełny,
    - d) drukarka odbiera właśnie znak,
    - e) trwa inicjalizacja drukarki
    - f) wystąpił inny błąd w pracy drukarki
  - PA2 do linii PAPEMPTY: wysoki poziom logiczny na tej linii oznacza wyczerpanie się papieru
  - PA3 do linii SELOUT: wysoki poziom logiczny na tej linii oznacza, że drukarka jest fizycznie połączona z komputerkiem – jest w stanie "on-line"
  - linie PA4...PA6 portu pozostają niewykorzystane.

Pozostało nam podłączenie trzech sygnałów sterujących drukarkę

- PC4 do linii AUTOFD: niski poziom podany przez 8255 na to wejście powoduje automatyczne dodawanie znaków w trybie tekstowym przez drukarkę po zakończeniu każdej linii
- PC5 do linii /RESET: niski poziom logiczny uruchamia w pewnych drukarkach procedurę inicjującą
- PC6 do linii SELIN: niski poziom logiczny na tej linii informuje drukarkę że jest ona wybrana i będzie używana. Zwykle linia ta jest połączona z masą, my jednak trzymając się zasady edukacyjnej naszego kursu umożliwimy sobie sterowanie tym sygnałem.

Gotowe! Teraz należy jeszcze dołączyć drukarkę za pomocą typowego kabla drukarkowego z gniazdem DB-25/F, które przed chwilą skrosowaliśmy z wyjściami portów PA, PB i PC układu 8255.

Zapiszmy teraz słowo konfiguracyjne rejestru CTRL układu 8255 dla trybu pracy z drukarką – czyli transmisji synchronicznej gdzie dane są wysyłane z 8255 do drukarki. I tak:

- bit 7 = 1 (zawsze jedynka)
- bit 6 i 5 = 00 (tryb pracy PA jako tryb 0)
- bit 4 = 1 (linie PA jako wejściowe będą odczytywać stany linii drukarki)
- bit 3 = 0 (linie PC5...PC7 będą sterowały sygnałami AUTOFD, SELIN i /RESET)
- bit 2 = 1 (tryb pracy rejestru PB – 1)
- bit 1 = 0 (PB jako wyjście)
- bit 0 = 1 (aby wolna linia PC3 na wszelki wypadek była wejściem, bo pozostałe linie PC0..PC2 generują sygnały synchronizacji transmisji a więc /ACK, /OBF i INTR)

Wobec tego liczba wpisana do rejestru CTRL przy inicjacji układu 8255 to będzie:

10010101 binarnie, czyli 95h szesnastkowo.

Wykonujemy więc instrukcję:

```
MOV A, #95h
MOV DPTR, #IO_CTRL
MOVX @DPTR, A
```

I gotowe, teraz należy jeszcze odblokować przerzutnik zezwalający na generowanie sygnału przerwania INTRB na linii PC0. Zgodnie z tym co opisałem wcześniej należy wpisać do rejestru CTRL układu 8255 bajt tak jak się ustawia pojedyncze bity rejestru PC. Jak wiemy przerzutnik INTRB znajduje się "pod bitem" D2 rejestru CTRL, dodatkowo ponieważ mamy zamiar ustawić ten bit aby odblokować działanie linii INTRB ustawiamy także bit 0, wobec tego otrzymamy wartość:

00000101 binarnie, czyli 05h szesnastkowo, zapisujemy ją do rejestru CTRL podobnie jak poprzednio:

```
MOV A, #05h
MOV DPTR, #IO_CTRL
MOVX @DPTR, A
Dodatkowo należy zainicjować sygnały sterujące drukarką, czyli linie PC5...PC7 w sposób następujący:
```

```
MOV A, #01010000b ;SELIN=1, RESET=0 (inicjacja), AUTOFD=1
MOV DPTR, #IO_PC
MOVX @DPTR, A
```

a po chwili powinniśmy wykonać instrukcje ustawiające linię /RESET drukarki w stanie nieaktywnym, czyli:

```
MOV A, #01110000b ;SELIN=1, RESET=1 (praca), AUTOFD=1
MOV DPTR, #IO_PC
MOVX @DPTR, A
```

i to już naprawdę wszystko, drukarka jest gotowa do przyjmowania i drukowania znaków.

Teraz trzeba wysłać znak a następnie czekać aż drukarka go wydrukuje i zgłosi ten fakt. Poniżej przedstawiam procedurę drukowania znaku na drukarce dołączonej do komputerka edukacyjnego. Drukowany jest znak z akumulatora Acc.  
*patrz tabela na następnej stronie*

Najpierw utworzyłem zbiór deklaracji adresów układu 8255 dołączonego do komputerka zgodnie z rys.2. Zbiór nazwałem "port8255.inc" i deklarację włączenia go do kompilacji umieściłem w zbiorze w procedurę PRNACC, która wysyła znak do drukarki. Jeżeli transmisja nie powiodła się to procedura wypisuje na displayu napis o błędzie "Err" oraz przed zakończeniem dodatkowo ustawia znacznik C, jeżeli wszystko przebiegło pomyślnie, to procedura kończy się z wyzerowanym znacznikiem C.

Uwaga, procedura ta sprawdza jedynie stan linii INTRB, nie sprawdza stanów linii informacyjnych drukarki /BUSY, PAPEMPTY i SELOUT. Proponuję jako zadanie domowe uzupełnić tę procedurę o detekcję

# Też to potrafisz

Listing 1

```

1          CPU '8052.DEF'
2

Zbiór: "const.inc"
Zbiór: "bios.inc"

Zbiór: "port8255.inc"
1          ;*****
2          ;Deklaracja adresów portów układu 8255
3          ;klasa mikroprocesorowa - odcinek 18, EdW 11/98
4          ;*****
5
6 A000     IO_PA         equ     A000h
7 A001     IO_PB         equ     A001h
8 A002     IO_PC         equ     A002h
9 A003     IO_CTRL       equ     A003h
10

Zbiór: "prnacc.s03"
5
6 0A00     org 0A00h          ;offset w obrębie pamięci EPROM komputera
7
8          ;*****
9          ;* PRNACC * Wysyła znak z akumulatora na drukarkę
10         ;*****
11 0A00 C083   prnAcc:   push    DPH
12 0A02 C082   push    DPL
13 0A04 C0E0   push    Acc
14 0A06 90A000 chkprn:   mov     DPTR,#IO_PA          ;odczyta stanu drukarki
15 0A09 E0     movx   A, DPTR
16 0A0A 540F   anl    A,#0Fh
17 0A0C 6409   xrl   A,#1001b          ;czy drukarka gotowa ?
18 0A0E 6017   jz     prnok           ;tak to drukuj
19 0A10         prnerror:
20 0A10 120274   lcall  CLS
21 0A13 757879   mov   DL1,#_E
22 0A16 757950   mov   DL2,#_r
23 0A19 757A50   mov   DL3,#_r
24 0A1C 1202C5   lcall CONIN          ;czekanie na dowolny klawisz
25 0A1F D3     setb  C              ;ustawienie znacznika błędu (C=1)
26 0A20 D0E0   pop   Acc
27 0A22 D082   exit:  pop   DPL          ;i zakończenie drukowania
28 0A24 D083   pop   DPH
29 0A26 22     ret
30 0A27 D0E0   prnok:  pop   Acc          ;od tej instrukcji gdy OK !
31 0A29 90A001   mov   DPTR,#IO_PB
32 0A2C F0     movx  DPTR,A          ;wysłanie znaku do drukarki
33 0A2D 90A000   mov   DPTR,#IO_PA    ;odczyt stanu drukarki
34 0A30 E0     wait:   movx  A, DPTR
35 0A31 5480   anl   A,#80h         ;czy można wysłać następny znak ?
36 0A33 60FB   jz    wait           ;nie to czekaj
37 0A35 C3     clr   C              ;zakoczenie drukowania OK. !
38 0A36 80EA   sjmp  exit
39
40 0A38     END

Kompilacja zakończona pomyślnie !
Zbiór: "prnacc.s03" , 56 bajt(ów), 0.1 sekund(y).
```

tych sygnałów i spróbować w domowym zaciszu pobawić się z tymi sygnałami.

Procedura została skompilowana z offsetem 0A00h, czyli w obrębie pamięci programu komputerka. Jeżeli ktoś chce, to może zmodyfikować pamięć monitora dodając na stałe procedurę PRNACC do pamięci programu EPROM. Sposób modyfikacji przedstawiłem we wrześniowym numerze EdW. W każdym razie trzeba dysponować programatorem pamięci EPROM.

Jeżeli nie posiadasz programatora to procedurę PRNACC powinieneś wykorzystywać w aplikacjach pisanych przez siebie, tzn. umieszczać listing podany powyżej w pliku źródłowym twego programu (oczywiście pomijając dodatkowe oznaczenia i informacje wygenerowane przez kompilator PASM51.EXE).

Użycie przedstawionej procedury jest bardzo proste i wyglądać może właśnie tak:

```

MOV   A, #A'          ;wydrukuj literę A
LCALL PRNACC
JNC  następny _znak
blad_drukarki:
.....
.....;instrukcje w wypadku błędu drukowania
```

Na deser proponuję krótką procedurę dodatkową, która korzysta z procedury PRNACC i drukuje tekst (ciąg bajtów) zakończony znakiem #0 (zero) zaczynający się od adresu podanego w DPTR. Oto ona:

Teraz możesz np. w swoim programie testującym spróbować wydrukować zdanie:

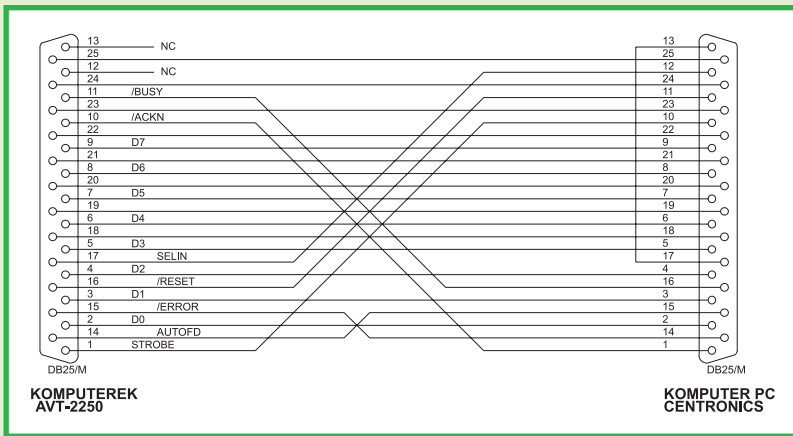
Listing 2

```

CPU '8052.def'
PRNACC equ 0A00h

;*****
;* PRNXT * Wysyła tekst ASCII na drukarkę (dane: mov DPTR,#tekst)
;*****
PRNXT:
movx A, DPTR          ;pobranie znaku z bufora
cjne A,#0,ok1        ;czy znak końca tekstu ?
ret                  ;tak to koniec procedury
ok1: lcall PRNACC      ;nie to wydrukuj znak
jnc  ok2             ;czy błąd drukowania ?
ret                  ;tak to zakończ procedurę
ok2: inc DPTR         ;nie to następny znak
sjmp prnxt
ret
```





Rys.8 Schemat kabla do połączenia komputera PC z AVT-2250

"Hello z systemu AVT-2250!" za pomocą sekwencji instrukcji:

```
MOV DPTR, #zdanie
LCALL PRNXTX
.....
.....
zdanie db 'Hello z systemu AVT-2250!', 0
```

Pamiętaj aby na końcu zawsze postawić znak #0, w przeciwnym przypadku procedura pójdzie w przysłowiowe "maliny".

## SYMULOWANIE DRUKARKI, CZYLI JAK ODBIERAĆ ZNAKI Z ZEWNĄTRZ

Uff, wiesz już drogi czytelniku w jaki sposób wysyłać znaki na drukarkę za pomocą układu 8255. Czy nie warto byłoby także móc ich odbierać, ale skąd?, no np. z zewnętrznego urządzenia wyposażonego w port równoległy kompatybilny z naszym, chociażby z pocziwego PC'ta lub innego komputera, jaki posiadasz.

W tej części artykułu pokażę w jaki sposób komputer edukacyjny AVT-2250 może bez żadnych przeróbek zmontowanego już układu 8255 (zgodnie z rys.2) udawać drukarkę i przyjmować znaki z zewnątrz. Jak się pewnie domyślasz do zrealizowania tego celu wykorzystam ten sam tryb pracy portu jak poprzednio (tryb 1) lecz port PB będzie pracował jako wejście danych. Sytuację tę możesz zobaczyć na rys. 4 w tej jego części gdzie pokazany jest port PB w trybie 1 oraz towarzyszące mu sygnały synchronizujące transmisję, czyli /STR, INB i INTRB.

Pierwszym krokiem jest ustalenie składni słowa konfiguracyjnego portu CTRL układu 8255. Wszystkie bitu tego słowa będą takie same jak poprzednio za wyjątkiem dwóch najmłodszych, oto one:

bit 1 = 1 ( PB jako wejście )  
bit 0 = 0 ( PCb jako wyjście )

Listing 3

```
7 0B00 org 0B00h
8
9 ;*****
10 ;* INPACC * Czekaj na znak z portu rownoleglego i do Acc
11 ;*****
12 0B00 C083 INPACC: push DPH
13 0B02 C082 push DPL
14 0B04 90A000 mov DPTR,#IO_PA ;czy układ 8255 zajety ?
15 0B07 E0 waitp: movx A, DPTR
16 0B08 5480 anl A,#80h
17 0B0A 60FB jz waitp ;tak to poczekaj
18 0B0C 90A001 mov DPTR,#IO_PB ;nie to odczytaj dana
19 0B0F E0 movx A, DPTR ;dana w Acc
20 0B10 D082 pop DPL
21 0B12 D083 pop DPH
22 0B14 22 ret
23
24 0B15 END
```

Kompilacja zakończona pomyslnie !  
Zbior: "inpacc.s03" , 21 bajt(ow), 0.1 sekund(y).

stąd więc otrzymamy słowo: 10010110 binarnie, czyli 96h szesnastkowo. Podobnie jak poprzednio trzeba wysłać jest do portu 8255, właśnie tak:

```
MOV A, #96h
MOV DPTR, #IO_CTRL
MOVX @DPTR, A
```

Pozostał jeszcze do odblokowania przerzutnik zgłoszenia przerwania INTRB

tak więc do portu układu 8255 wysyłamy taką samą wartość jak w przypadku obsługi drukarki, czyli:

```
MOV A, #05h
MOV DPTR, #IO_CTRL
MOVX @DPTR, A
```

Następnie należy wykonać specjalny kabel do połączenia z urządzeniem zewnętrznym, w naszym przypadku będzie to komputer PC z równoległym złączem typu Centronics (drukarkowym). Rys.8 przedstawia schemat połączeń kabla do transmisji z komputerem PC. Oba wtyki po dwóch stronach kabla będą typu męskiego DB-25/M, lecz ze względu na wykonane połączenia należy zaznaczyć (flamastrem) tę stronę którą wtykamy do komputera PC a drugą, jako dołączaną do portu 8255 komputerka AVT-2250. Podczas wykonywania kabla radzę Ci zastanowić się na przyporządkowaniu poszczególnych linii sygnałom portu Centronics komputera PC. Dla ułatwienia powiem tylko że dzięki skrzyżowaniom sygnałów możliwe jest połączenia następujących kompatybilnych par sygnałów:

AVT-2250	↔	PC
STROBE	z	ACK
BUSY	z	RESET
ERROR	z	AUTOFD

Jeżeli kabel został wykonany to można zabrać się do zainicjowania stanów na odpowiednich liniach łącza, tak aby komputer PC zaczął "widzieć" nasz komputer i aby można było wysłać do niego jakieś znaki.

W tym celu do portu PC należy wysłać bajt: 00100000b, a to dlatego, że wtedy ustawimy sygnały:

PAPER EMPTY (OK.), BUSY i ERROR, tak że komputer będzie widziany przez PC'ta jak potencjalna gotowa do pracy drukarka. Wykonujemy więc instrukcje:

```
MOV A, #00100000b ;PAPER OK, BUSY, not ERROR
MOV DPTR, #IO_PC
MOVX @DPTR, A
```

i układ jest gotowy do odbierania znaków z urządzenia zewnętrznego.

Poniżej zamieszczam uproszczoną procedurę oczekiwania na znak z łącza równoległego i umieszczenie go w akumulatorze. Jako pracę domową proponuję uzupełnienie tej procedury funkcją przeterminowania, tak aby komputer edukacyjny przy braku transmisji po prostu nie zawiesił się.  
patrz tabela na następnej stronie

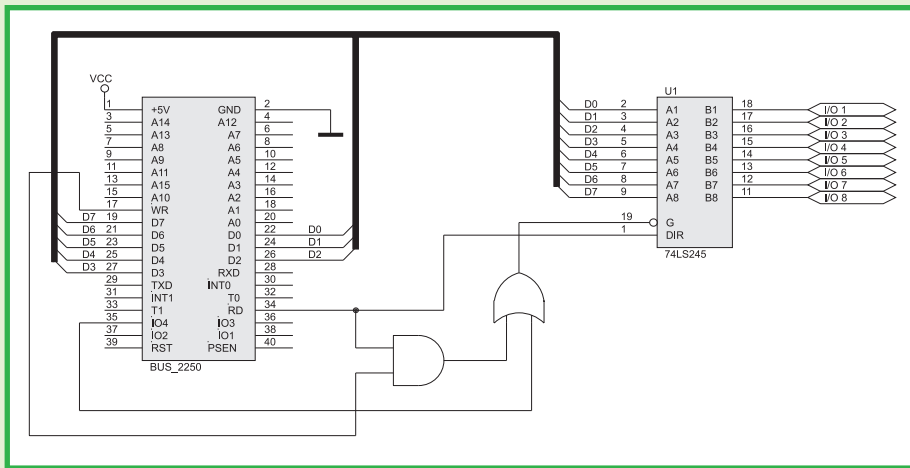
Procedura została skompilowana z off-setem 0B00h, tak aby znaleźć się w obszarze pamięci programu monitora komputerka, a jednocześnie nie kolidować z wcześniejszą procedurą obsługi drukarki. Tak ja poprzednio procedury tej można także używać w swoich aplikacjach wstawiając ją do zbioru źródłowego.

Korzystanie z procedury INPACC jest proste i sprowadza się do jej wywołania:

```
LCALL INPACC
.....
```

;a co dalej zrobisz z tym znakiem to twoja sprawa

# Też to potrafisz



Rys.1 Rozwiązanie zadania nr 1

To na razie wszystko, co przygotowałem na dzisiejszy odcinek klasy mikroprocesorowej. Myślę że spora ilość informacji do przemyślenia a także sprawdzenia w praktyce umili Ci, drogi Czytelniku jesienne chłodne wieczory. Powodzenia.

Stawomir Surowiński

## LEKCJA 11

Rozwiązania zadań z poprzedniej lekcji przedstawione są na rysunkach poniżej. I tak rysunek 1 przedstawia zmodyfikowany z rys.9 w poprzednim numerze układ uwzględniający sygnał /WR procesora.

Na rys.2 pokazałem jak połączyć dwa układy 74LS138 tak aby zastąpić jeden demultiplekser 74LS154.

Wreszcie na rys.3 znajduje się zmodyfikowany układ dekodera z rys.10 (z poprzedniego numeru EdW), dzięki któremu układ U1 zapisywany jest pod adresem A000h a odczytywany pod adresem A001h

Sądzę że analiza powyższych schematów nie nastręczy nikomu trudności. Jako zadanie na kolejny raz proponuję zapisanie procedury drukowania zawartości zewnętrznej pamięci danych komputerka AVT-2250, w zadany z klawiatury zakresie np. od adresu 8000h do 80FFh w postaci szesnastkowej i znakowej według przykładowego szablonu:

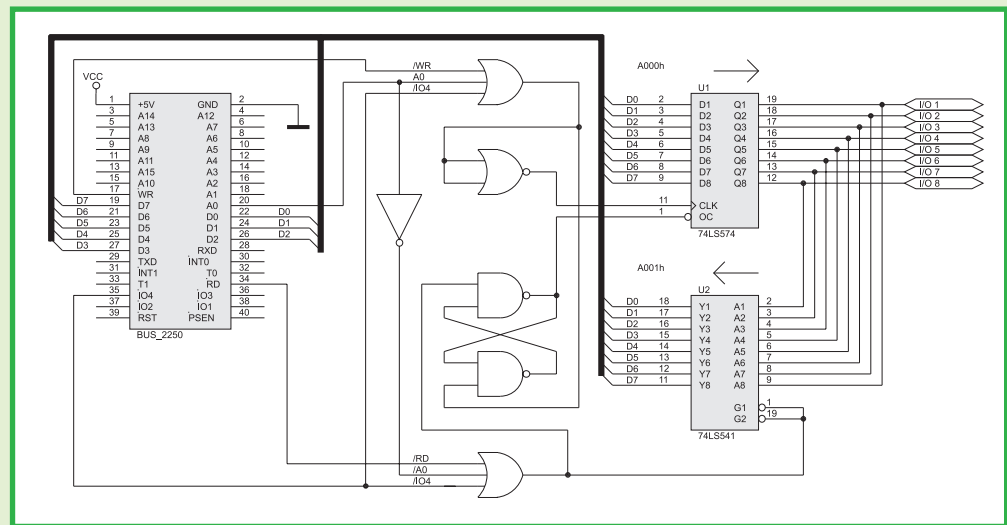
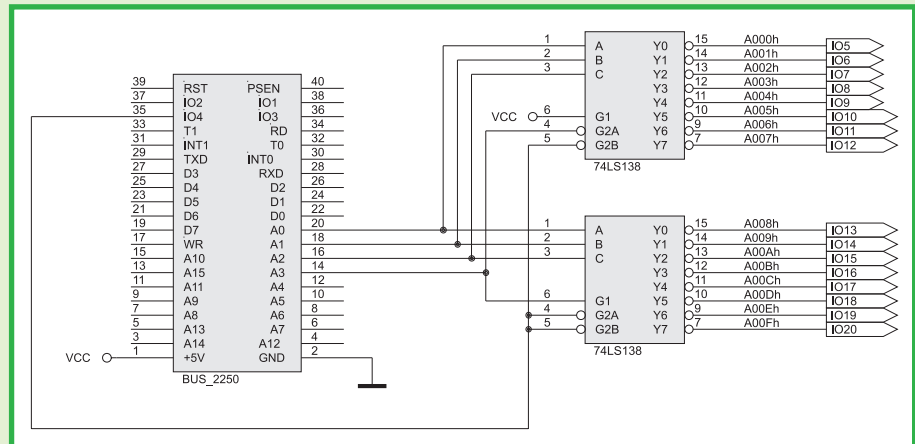
Adres - 16 bajtów kolejnych danych w HEX - te same bajty jako znaki

Przykładowy wydruk powinien wyglądać jak ten na dole strony

Wydruk przedstawia zawartość pamięci o podanych adresach z zakresu: 2600h... 26FFh. Wesołej zabawy i do zobaczenia w następnym numerze EdW, gdzie znajdzie się rozwiązanie powyższego zadania.

Stawomir Surowiński

Rys.2 Rozwiązanie zadania nr 2



Rys.3 Rozwiązanie zadania nr 3

Address	Data	Character
(2600h)	0E 57 31 C0 50 9A E2 36 7D 0E 9A DF 35 7D 0E 80	.W1.P.6]...5].
(2610h)	3E CA 8A 00 74 06 C6 46 FB 01 EB 04 C6 46 FB 02	>...t.F....F.
(2620h)	8A 46 FB 50 B0 0F 50 9A AA 10 CF 0C 8D 7E FB 16	.FP.P.....~.
(2630h)	57 B0 02 50 E8 05 F8 8D 7E FC 16 57 9A 1F 0C CF	W..P...~.W....
(2640h)	0C 80 3E 7E 89 00 74 07 C6 06 7E 89 00 EB 17 8A	...>...t...~....
(2650h)	46 FB 3C 01 75 07 C6 06 CA 8A 01 EB 09 3C 02 75	F.<...t.....<.u
(2660h)	05 C6 06 CA 8A 00 89 EC 5D C3 55 89 E5 31 C0 9A	.....].U..1..
(2670h)	30 05 7D 0E BF 98 89 1E 57 B8 B7 00 50 9A E3 37	0.).....W..P..7
(2680h)	7D 0E BF 98 89 1E 57 BF 18 8A 1E 57 9A 96 38 7D	].....W....W..8]
(2690h)	0E 83 C4 04 BF 98 89 1E 57 9A 5B 38 7D 0E 9A 47	.....W.[8]..G
(26A0h)	0C CF 0C B0 07 50 E8 8F FA 31 C0 9A 16 01 7D 0E	.....P..1.....].
(26B0h)	5D C3 00 55 89 E5 31 C0 9A 30 05 7D 0E 80 3E C8	]..U..1..0.]...>
(26C0h)	8A 00 B0 00 75 01 40 A2 C8 8A BF 94 89 1E 57 9A	.....U.....W.
(26D0h)	1F 0C CF 0C C6 06 7B 89 00 C6 06 7E 89 01 C6 06	.....f.....~....
(26E0h)	52 90 FF B0 00 50 BF 32 0C 0E 57 E8 7C FB 5D C3	R....P2..W1.]..
(26F0h)	55 89 E5 31 C0 9A 30 05 7D 0E A0 CB 8A 30 E4 48	U..1..0.]...0.H